**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**TÊN ĐỀ TÀI: PUZZLE GAME**

**SVTH : ĐINH MINH THIỆN**

**MSSV : 18110205**

**SVTH : NGUYỄN GIA HUY**

**MSSV : 18110121**

**TP. Hồ Chí Minh, tháng 11, năm 2019**

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc27208354)

[I. PHẦN MỞ ĐẦU 2](#_Toc27208355)

[1. Lí do chọn đề tài 2](#_Toc27208356)

[2. Mục tiêu cho đề tài 2](#_Toc27208357)

[3. Phương pháp nghiên cứu 2](#_Toc27208358)

[4. Giới thiệu đề tài 2](#_Toc27208359)

[II. PHẦN NỘI DUNG 4](#_Toc27208360)

[1. Mô tả ứng dụng 4](#_Toc27208361)

[1.1. Cơ sở lý thuyết 4](#_Toc27208362)

[1.2. Chức năng phần mềm 4](#_Toc27208363)

[1.3. Các hàm và biến chính của phần mềm 4](#_Toc27208364)

[2. Mô tả quá trình thực hiện 5](#_Toc27208365)

[2.1. Thiết kế giao diện 5](#_Toc27208366)

[2.2. Ứng dụng kiểu dữ liệu Stack trong Trò chơi Xếp hình 7](#_Toc27208367)

[2.3. Viết code phần xử lí 8](#_Toc27208368)

[2.4. Chạy và kiểm thử 11](#_Toc27208369)

[2.5. Sửa các lỗi gặp phải 13](#_Toc27208370)

[KẾT LUẬN 14](#_Toc27208371)

[TÀI LIỆU THAM KHẢO](#_Toc27208372)

[PHỤ LỤC: BẢNG KẾ HOẠCH](#_Toc27208373)

**MỤC LỤC HÌNH**

[Hình 1: Mô hình trò chơi Xếp hình trong thực tế. 3](#_Toc27208332)

[Hình 2: Hình ảnh các khối hình được chuẩn bị. 5](#_Toc27208333)

[Hình 3: Giao diện cơ bản sau khi thiết kế. 6](#_Toc27208334)

[Hình 4: Các Stack được sử dụng trong đề tài. 7](#_Toc27208335)

[Hình 5: Hàm xử lí nút “Shuffle”. 8](#_Toc27208336)

[Hình 6: Hàm xử lí Nút “Undo”. 9](#_Toc27208337)

[Hình 7: Hàm xử lí Swap 10](file:///C:\Users\PC\Desktop\baocao.docx#_Toc27208338)

[Hình 8: Hàm kiểm tra Win. 11](#_Toc27208339)

[Hình 9: Trò chơi Xếp hình khi vừa mở lên. 11](#_Toc27208340)

[Hình 10: Tiến hành di chuyển khối ảnh. 12](#_Toc27208341)

[Hình 11: Tiến hành Undo. 12](#_Toc27208342)

[Hình 12: Thông báo chúc mừng khi người chơi hoàn thành trò chơi. 13](#_Toc27208343)

[Hình 13: Hàm xử lí trộn. 14](#_Toc27208344)

# LỜI NÓI ĐẦU

Cấu trúc dữ liệu và giải thuật là một trong những môn học nền tảng quan trọng nhất dành cho sinh viên ngành Công nghệ Thông tin. Môn học cung cấp cho sinh viên hiểu và biết cách sử dụng các kiểu dữ liệu cơ bản thường gặp trong quá học tập và làm việc. Giúp người học biết được sự tác động lẫn nhau giữa dữ liệu và giải thuật để người sử dụng có thể giải quyết các vấn đề gặp phải trong bài toán và thực tế.

Sau quá trình học tác giả đã vận dụng kiến thức tiếp thu được để ứng dụng thực hiện đồ án môn học với đề tài: Puzzle Game.

# PHẦN MỞ ĐẦU

## Lí do chọn đề tài

Với đề tài “Puzzle Game” có thể giúp tác giả vận dụng những kiến thức học được từ môn học, đặc biệt là hiểu rõ hơn về kiểu dữ liệu trừu trượng Stack. Ngoài ra việc tạo ứng dụng Puzzle Game sẽ giúp người chơi có thể chơi được trò chơi mà không cần mô hình thực tế, phục vụ cho nhu cầu giải trí của người chơi.

## Mục tiêu cho đề tài

Hoàn thành sản phẩm, có thể cho người chơi được các thao tác (chơi và chơi lại, di chuyển các hình, sử dụng chức năng undo ), trải nghiệm trò chơi thành công. Giúp bản thân tác giả hiểu được, hiểu sâu hơn về kiểu dữ liệu trừu tượng ngăn xếp (Stack). Có thể sử dụng thành thạo các phương thức và thuộc tính của Stack.

## Phương pháp nghiên cứu

Vận dụng các kiến thức đã học được từ bộ môn “Cấu trúc dữ liệu và giải thuật”, tham khảo các nguồn tài liệu, giáo trình được cung cấp. Ngoài ra còn tham khảo thêm các nguồn tài liệu khác trên mạng internet và vận dụng tất cả để thực hiện đề tài nghiên cứu một cách thành công nhất.

## Giới thiệu đề tài

Puzzle Game (trò chơi giải đố) từ lâu đã trở thành một loại trò chơi giải trí được ưa chuộng trên toàn thế giới do tính giải trí cao, dễ tiếp cận và niềm vui mà nó mang lại cho người chơi khi họ hoàn thành câu đó mà trò chơi mang lại, puzzle game là cách gọi dành cho rất nhiều trò chơi có đặc điểm chung về tính giải đố, thử thách tư duy người chơi với trò xếp hình là một trong số các trò chơi như vậy

Về luật chơi người chơi được cho 1 bức ảnh bị cắt thành nhiều phần bằng nhau và được sắp xếp lộn xộn với một góc trống, người chơi phải xếp lại thành bức ảnh hoàn chỉnh theo các quy tắc:

* Chỉ được di chuyển các mảnh bao quanh ô trống
* Chỉ được di chuyển lên, xuống hoặc qua trái, qua phải
* Mỗi lần di chuyển chỉ được di chuyển 1 ô và sẽ được tính là 1 bước



##### Hình 1: Mô hình trò chơi Xếp hình trong thực tế.

# PHẦN NỘI DUNG

## Mô tả ứng dụng

### Cơ sở lý thuyết

Ngăn xếp (Stack) là một dạng danh sách được cài đặt nhằm sử dụng cho các ứng dụng cần xử lý theo thứ tự đảo ngược. Trong cấu trúc dữ liệu ngăn xếp, tất cả các thao tác thêm, xóa một phần tử đều phải thực hiện ở đầu một danh sách, đầu này gọi là đỉnh (top) của ngăn xếp. Có thể hình dung ngăn xếp thông qua hình ảnh một chồng đĩa đặt trên bàn. Nếu muốn thêm vào một đĩa, người ta phải đặt nó lên trên đỉnh. Nếu lấy ra một đĩa, người ta cũng phải lấy đĩa trên đỉnh chồng. Ngăn xếp là một cấu trúc trừu tượng có tính chất “*vào sau ra trước”* (Last In First Out - LIFO) hay “*vào trước ra sau”* (First In Last Out – FILO) [1]­

### Chức năng phần mềm

Mô phỏng trò chơi Xếp hình thành ứng dụng để người chơi có thể chơi mà không cần có mô hình thực tế. Ứng dụng trong giải trí, rèn luyện tư duy logic, trong học tập và giảng dạy về ngăn xếp và lập trình căn bản.

### Các hàm và biến chính của phần mềm

#### Hàm Shuffle

* Mục đích: trộn hình.
* Tình huống sử dụng: khi người chơi muốn chơi lại.

#### Hàm Moveset

* Mục đích: các thao tác dùng để trộn hình.
* Tình huống sử dụng: mỗi khi hàm Shuffle được gọi.

#### Biến move:int

* Mục đích: đếm số bước để hiển thị ra giao điện.
* Tình huống sử dụng: khi người chơi di chuyển 1 khối ảnh.

#### Hàm Swap

* Mục đích: thực hiện việc đổi chỗ khối ảnh và khối trống.
* Tình huống sử dụng: giúp người chơi di chuyển khối ảnh sang vị trí trống.

#### Hàm Win

* Mục đích: hàm điều kiện của hệ thống để kiểm tra xem liệu người chơi đã hoàn thành trò chơi.
* Tình huống sử dụng: mỗi khi hàm Swap được gọi.

#### Hàm Undo

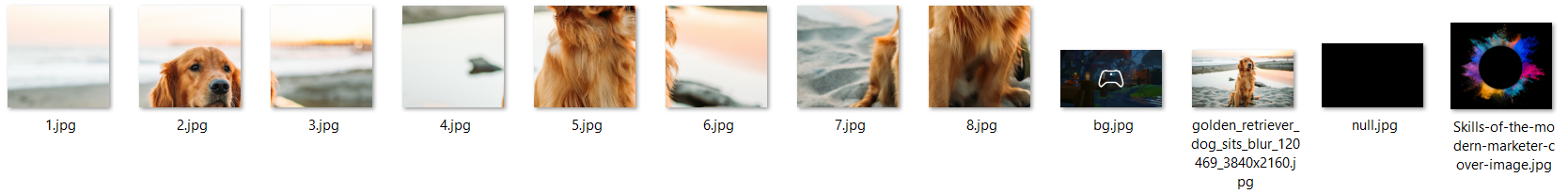
* Mục đích: cho người sử dụng khi muốn quay lại bước trước đó.
* Tình huống sử dụng: khi người chơi muốn quay lại bước trước đó.

## Mô tả quá trình thực hiện

### Thiết kế giao diện

Việc thiết kế giao diện của ứng dụng là một phần rất quan trọng trong việc xây dựng ứng dụng, là bộ mặt của ứng dụng, nơi người dùng trực tiếp thao tác với ứng dụng. Tạo sự hứng thú, thu hút cho người chơi.

#### Chuẩn bị hình ảnh

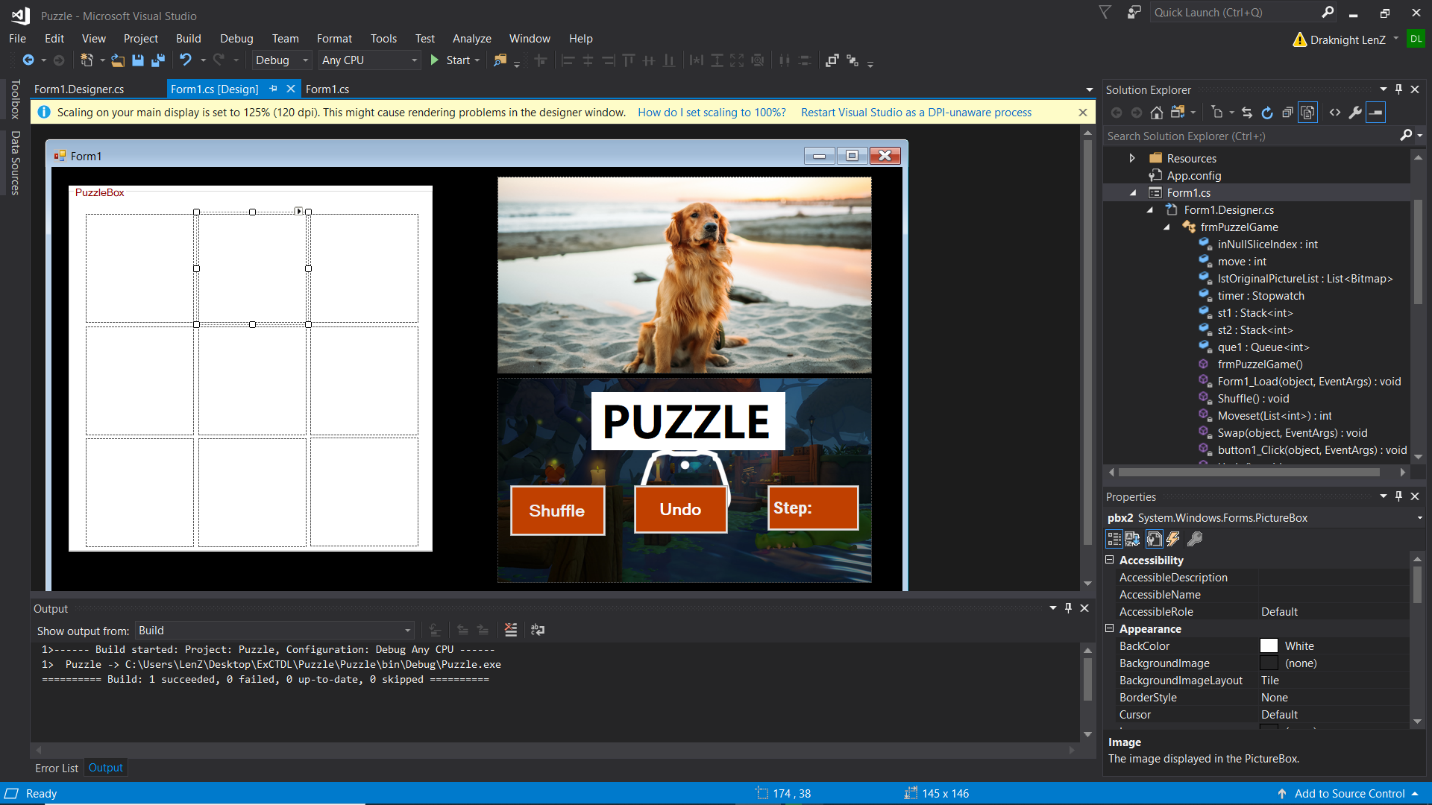


##### Hình 2: Hình ảnh các khối hình được chuẩn bị.

* Tùy chỉnh size ảnh phù hợp: đây là một phần rất quan trọng, các khối phải có cùng chiều cao và size chiều rộng (cụ thể 145x146). Việc tạo size ảnh phù hợp sẽ giúp việc lập trình các hàm sử lí được đơn giản, có tính thống nhất và dễ dàng hơn rất nhiều vì thế chúng ta nên tùy chỉnh một cách phù hợp nhất có thể.

#### Thiết kế giao diện với Windows Forms Application

Kể từ lúc này chúng ta sẽ bắt đầu sử dụng Windows Forms Application để thiết kế và lập trình ứng dụng.

* Sau khi tạo project thành công, chúng ta bắt đầu điều chỉnh form với các thao tác trên thanh properties. Chúng ta sẽ thay đổi màu nền (*Backcolor*), cố định size không cho tùy chỉnh (*MaximizeBoxFalse*), chỉnh lại mục *Name* và mục *Text*.
* Thêm hình ảnh khối hình, khối trống và hình mẫu từ thư mục vào *Project Resources*.
* Tạo các *PictureBox* trong thanh *Toolbox* để hiển thị các khối (các khối sẽ hiển thị ngay khi bắt đầu trò chơi nên ta điều chỉnh thuộc tính *Visible: True*) sau đó tạo thêm *label* ghi chú thứ tự các khối.
* Tiến hành tinh chỉnh vị trí, canh khoảng cách giữa các khối ảnh.
* Thêm ảnh mẫu và thiết kế khu vực đặt các nút điều khiển cũng như thông báo
* Tạo 3 nút (*Button*) “Shufle”, “Undo” và “Step: ”, căn giữa điều chỉnh vị trí phù hợp so với form.

##### Hình 3: Giao diện cơ bản sau khi thiết kế.

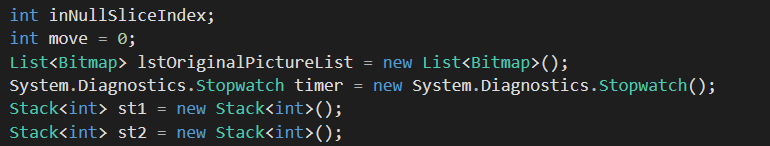
### Ứng dụng kiểu dữ liệu Stack trong Trò chơi Xếp hình

Ngăn xếp (Stack) là một dạng danh sách được cài đặt nhằm sử dụng cho các ứng dụng cần xử lý theo thứ tự đảo ngược. Trong cấu trúc dữ liệu ngăn xếp, tất cả các thao tác thêm, xóa một phần tử đều phải thực hiện ở đầu một danh sách, đầu này gọi là đỉnh (top) của ngăn xếp. Có thể hình dung ngăn xếp thông qua hình ảnh một chồng đĩa đặt trên bàn. Nếu muốn thêm vào một đĩa, người ta phải đặt nó lên trên đỉnh. Nếu lấy ra một đĩa, người ta cũng phải lấy đĩa trên đỉnh chồng. Ngăn xếp là một cấu trúc trừu tượng có tính chất “*vào sau ra trước”* (Last In First Out - LIFO) hay “*vào trước ra sau”* (First In Last Out – FILO) [1]­ . Dựa vào đặc tính này của ngăn xếp ta có thể ứng dụng ngăn xếp để lưu lại thông tin vị trí trước mỗi bước di chuyển của khối trống từ đó tạo ra chức năng Undo cho trò chơi

#### Các phương thức được sử dụng với kiểu dữ liệu Ngăn xếp (Stack) trong đề tài:

* *Push()*: thêm vị trí hiện tại vào *Stack*.
* *Pop():* lấy vị tri trước đó ra khỏi *Stack*.
* *Count:* kiểm tra xem *Stack* đang có bao nhiêu phần tử.

#### Các *Stack* trong đề tài:



##### Hình 4: Các Stack được sử dụng trong đề tài.

* Các *Stack* chứa vị trí lưu ở dạng *int.*

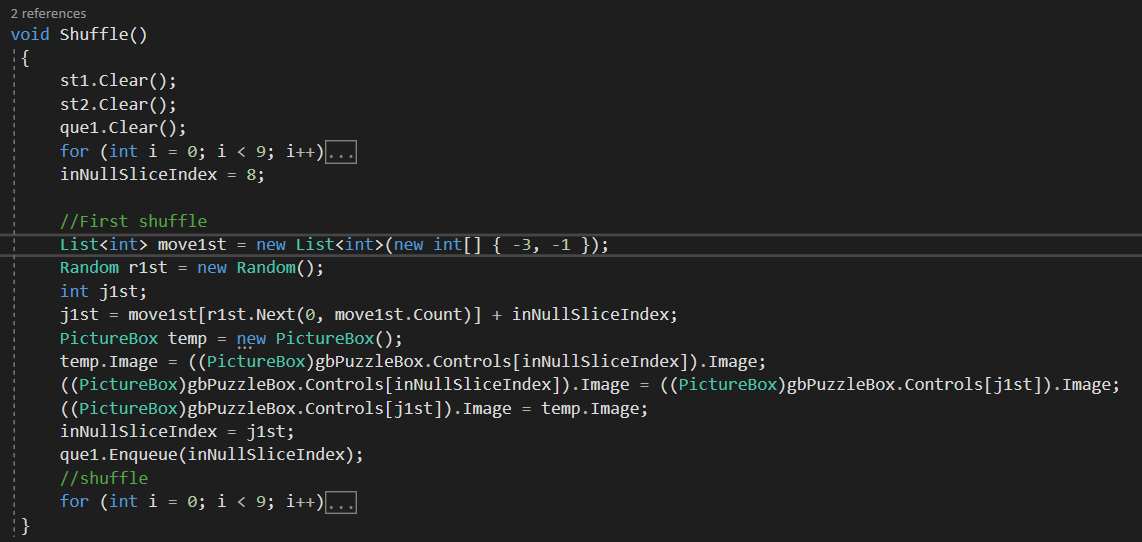
#### Sử dụng các Stack trong đề tài:

* Trong hàm xử lý *Swap*, mỗi khi hàm này được gọi, các *Stack* st1 và st2 sẽ lần lượt được gọi kèm để cập nhật lần lượt vị trí trước *Swap* và vị trí sau *Swap* của khối trống bằng hàm Push().
* Trong hàm cử lý *Undo*, ta kiểm tra xem liệu 2 *Stack* st1 và st2 có đồng thời khác 0 hay không bằng hàm(if (st1.Count != 0 && st2.Count != 0)) nếu cả 2 cùng khác không thì t dùng hàm Pop() để lấy dữ liệu vị trí trước đó và hiện tại lần lượt từ st1 và st2 sau đó thực hiện hoán đổi vị trí các khối ảnh, trường hợp ngược lại ta xuất ra thông báo “*Không thể Undo”.*

### Viết code phần xử lí

#### Nút “Shuffle”

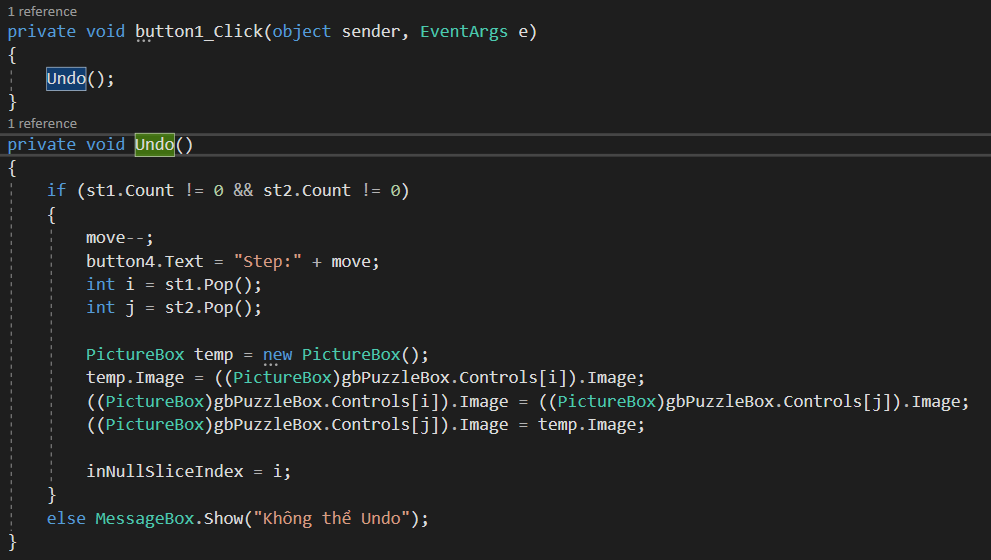
* Khi người chơi click vào nút “Shuffle”, hàm Shuffle sẽ được gọi cùng hàm Moveset để trộn các khối hình
* Đầu tiên 2 *Stack* st1 và st2 sẽ bị xóa bởi hàm Clear().
* Tiếp theo sẽ là lần trộn đầu tiên
* Vị trí khối trống sau lần trộn đầu sẽ được lưu vào 1 *hàng đợi*, *hàng đợi* này có nhiệm vụ lưu lại lần lượt các vị trí của khối trống sau mỗi lần trộn để đảm bảo với các lần trộn kế tiếp khối trống sẽ không quay về vị trí trước đó của nó



##### Hình 5: Hàm xử lí nút “Shuffle”.

#### Nút “Undo”

Với nút “Undo” có trường hợp để xử lí. Một là khi trò chơi đang diễn ra và người chơi chưa undo hết số lần thì nút này sẽ giúp trả lại hiện trạng các khối trước đó 1 bước đồng thời giảm số lượng “Step” đi 1, trường hợp 2 khi trò chơi chưa bắt đầu hoặc khi người chơi đã undo hết số lần thì sẽ hiện ra thông báo “Không thề Undo”.

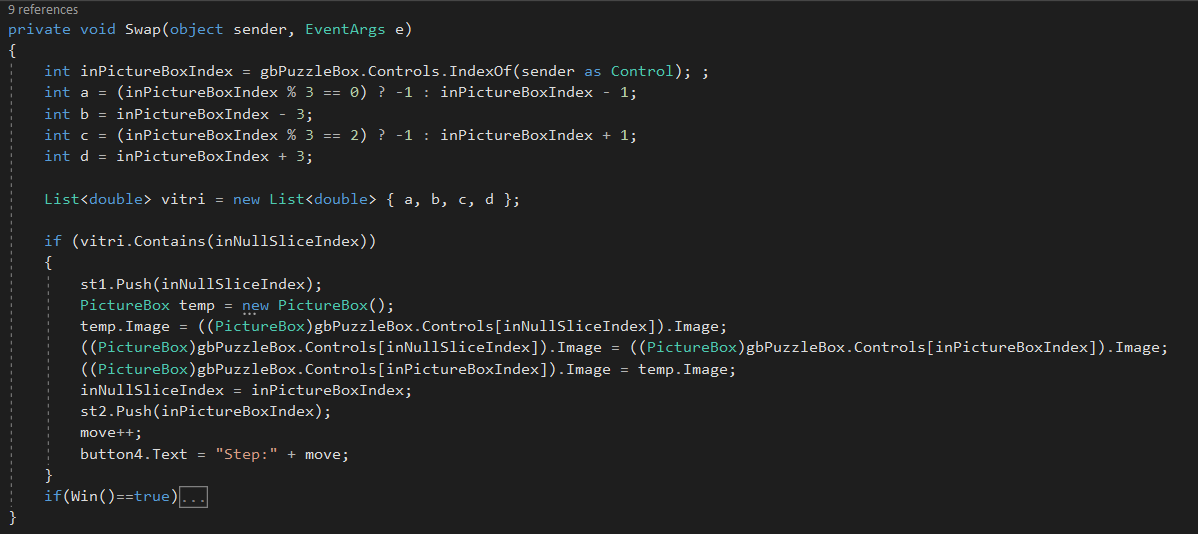


##### Hình 6: Hàm xử lí Nút “Undo”.

* Trường hợp 1 khi trò chơi còn diễn ra và người chơi chưa undo hết số lần: gọi lệnh *move--* để giảm “Step” xuống 1 đơn vị, gọi hàm Pop() để lấy ra vị trí hiện tại và vị trí trước đó từ 2 *Stack* st1 và st2 lưu vào biến i và j, thực hiện hoán đổi vị trí các khối có vị trí tương ứng i và j, cuối cùng cập nhật lại vị trí ô trống bằng hàm *inNullSliceIndex = i.*
* Trường hợp còn lại ta xuất ra thông báo “Không thể Undo”*.*

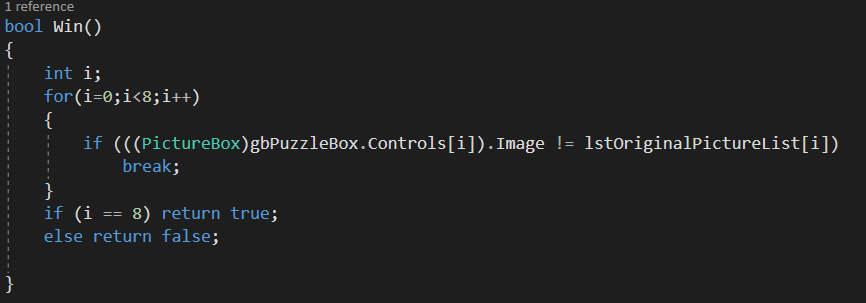
#### Hàm xử lí Swap khi click vào khối hình

Khi click vào hình sẽ được chia là 2 trường hợp. Trường hợp đầu là khi click vào khối xung quanh khối trống (trên, dưới, trái, phải) thì sẽ thực hiện Swap, trường hợp còn lại thì không làm gì cả

* Trường hợp 1 khi click vào khối xung quanh khối trống (trên, dưới, trái, phải), lệnh *st1.Push(inNullSliceIndex)* được gọi để lưu vị trí hiện tại của khối trống vào st1, hoán đổi vị trí của khối được chọn và khối trống sau đó lưu lại vị trí mới vào st2 bàng lệnh *st2.Push(inNullSliceIndex)*, tăng biến move rồi hiển thị số “Step” ra màng hình, sau đó gọi hàm “Win” để kiểm tra xem đã thắng hay chưa, nếu thắng thì xuất ra thông báo “YOU WIN !!!!!” và số bước, ngược lại thì không làm gì cả.
* Trường hợp còn lại thì không làm gì cả*.*

##### Hình 7: Hàm xử lí Swap

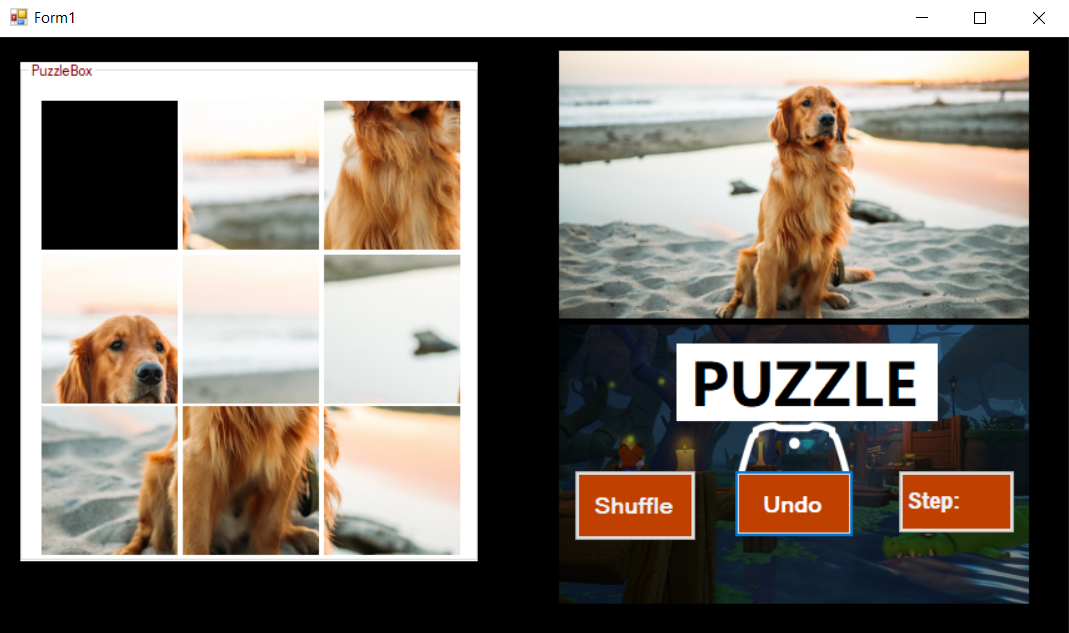
#### Hàm kiểm tra Win

Kiểm tra bằng cách chạy vòng lặp, nếu tất cả các khối đều đã đúng vị trí thì trà về True ngược lại thì trả về False

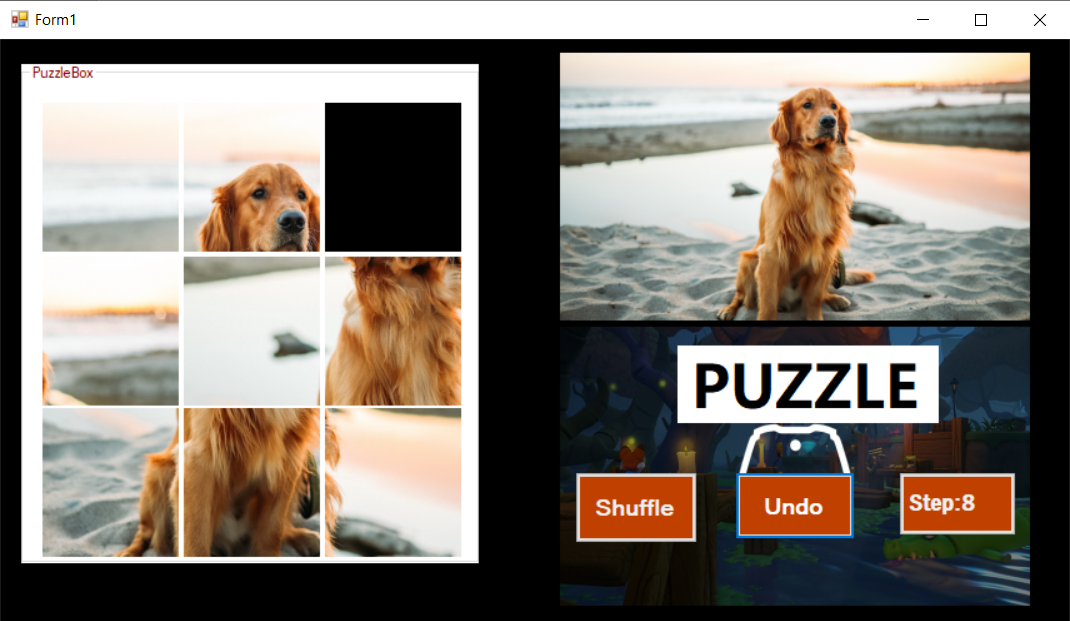
##### Hình 8: Hàm kiểm tra Win.

### Chạy và kiểm thử

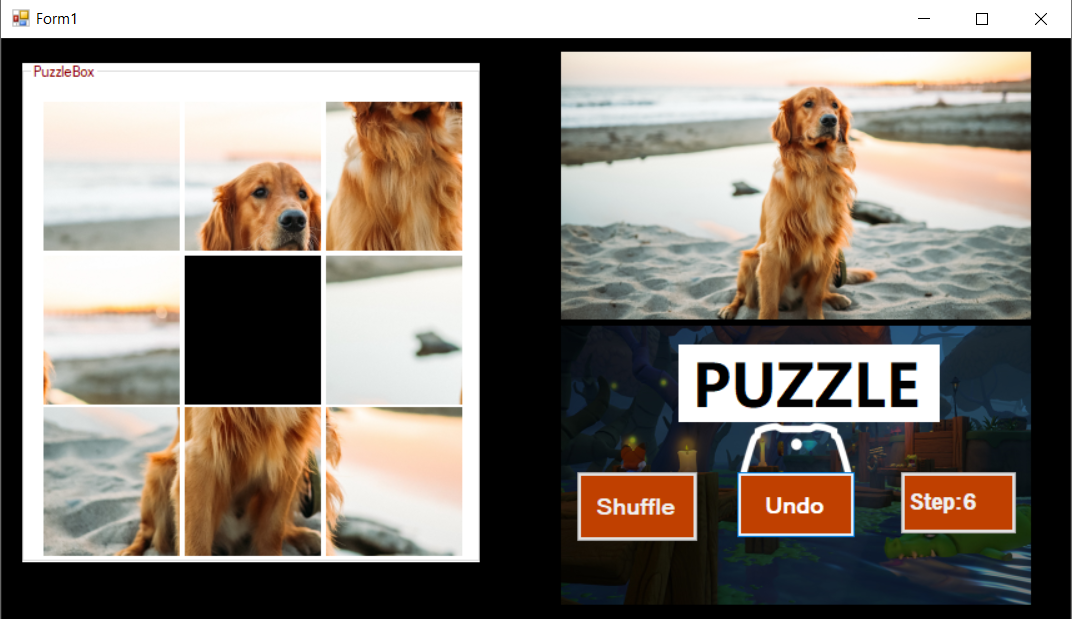
Mở chạy thử chương trình bắt đầu chơi thử.



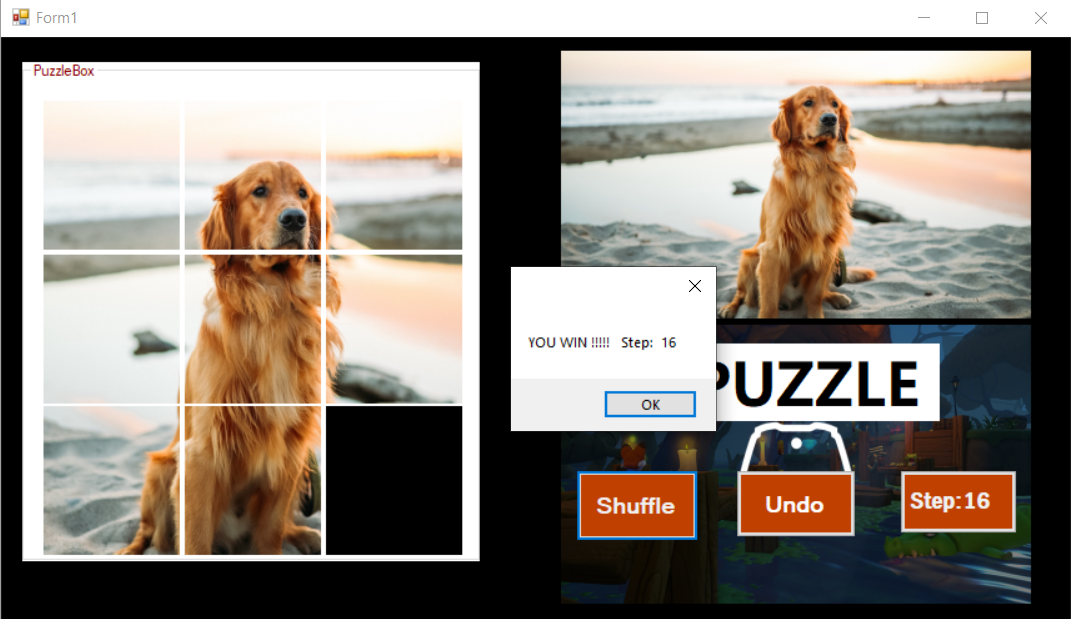
##### Hình 9: Trò chơi Xếp hình khi vừa mở lên.



##### Hình 10: Tiến hành di chuyển khối ảnh.



##### Hình 11: Tiến hành Undo.

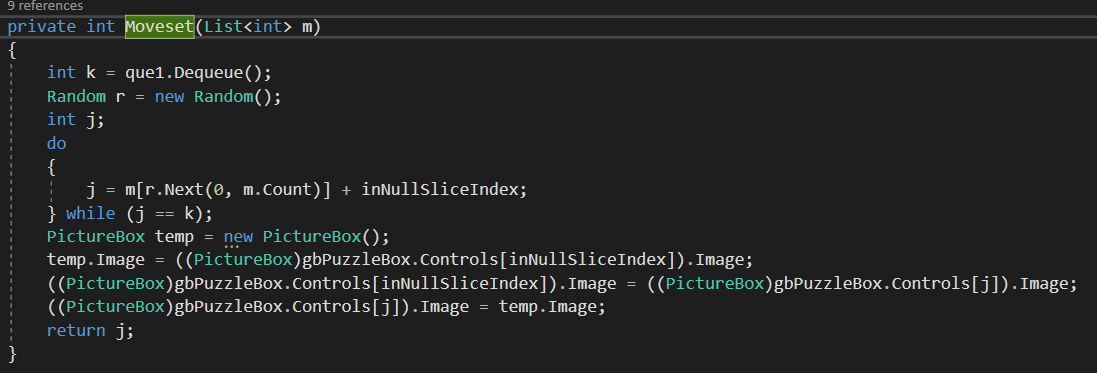


##### Hình 12: Thông báo chúc mừng khi người chơi hoàn thành trò chơi.

Tuy nhiên trong quá trình chạy thử đôi lúc đã xảy ra lỗi khi nhấn Shuffle thì không được trộn. Tiếp theo chúng ta sẽ tìm hiểu và sửa lỗi.

### Sửa các lỗi gặp phải

Như đã nêu ở trên trong quá trình chạy thử đã xảy ra lỗi, khi nhấn nút “Shuffle” thì hình không được trộn, lý do có thể là do trong quá trình trộn khối trống đã quay trở lại vị trí trước đó của nó, giải pháp đưa ra là thêm 1 *hàng đợi* có tác dụng lưu lại vị trí trước đó cùa khối trống, khi trộn thì dùng nó để kiểm tra ngăn không cho khối trống quay lại vị trí trước đó trong lúc trộn



##### Hình 13: Hàm xử lí trộn.

# KẾT LUẬN

* Mức độ hoàn thành mục tiêu: Đã tạo ra được 1 puzzle game (trò xếp hình) hoàn chỉnh và có thể chơi được, thành công ứng dụng lý thuyết trừu tượng Stack vào việc tạo ra tính năng undo trong trò chơi.
* Các khó khăn gặp phải: sử dụng chưa thành thạo Winform, việc tạo chuẩn bị hình ảnh để đưa vào giao diện chưa phù hợp dẫn đến phải sửa nhiều lần mới có size ảnh phù hợp.
* Cách khắc phục: tìm hiểu và học cách sử dụng Winform từ nhiều nguồn tài liệu trên internet, điều chỉnh size ảnh thích hợp cho việc code cũng như dễ nhìn cho giao diện.
* Ưu điểm của đồ án: sử dụng, khai thác hết các thuộc tính của kiểu dữ liệu Ngăn xếp (Stack), có tính ứng dụng trong giải trí và cả giáo dục phát triển tư duy logic, giao diện dễ nhìn, thân thiện, người chơi có thể dễ dàng thực hiện trò chơi.
* Những khuyết điểm của đồ án: so với các sản phẩm tương tự khác bên ngoài thì sản phẩm hiện tại vẫn còn chưa phát triển hoàn thiện nhất về các tính năng (như là chưa cho phép người dùng chọn độ khó trò chơi), giao diện chưa thật sự quá bắt mắt, cầu kì bằng các sản phẩm khác.
* Hướng phát triển đồ án: có thể thêm các tính năng khác như thêm một label hiển thị số lần di chuyển ít nhất để chiến thắng, quy định số lần di chuyển nhiều nhất,
* nếu vượt quá sẽ thông báo người chơi thua, làm thêm nhiều level khác dành cho những người chơi muốn thêm sự thử thách. Ngoài ra chúng ta có thể thêm âm thanh cho mỗi lần click chuột (hợp lệ và không hợp lệ sẽ có âm thanh khác nhau) và cả khi chiến thắng hay thua cuộc. Đầu tư thêm về phần xử lí đồ họa giúp tăng độ thẩm mỹ, bắt mắt thu hút người chơi hơn.

# TÀI LIỆU THAM KHẢO

[1] “GIÁO TRÌNH: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT” của tác giả: Lê Văn Vinh, NXB Đại học Quốc gia TP. Hồ Chí Minh.

Tài liệu tham khảo cho đề tài:

* Video hướng dẫn “**How to Create Picture Puzzle Game in C#”** của CodAffection
* Bài viết “**Lập trình Winform cơ bản”** của trang howkteam.vn

# PHỤ LỤC: BẢNG KẾ HOẠCH

|  |  |  |  |
| --- | --- | --- | --- |
| **DANH SÁCH CÔNG VIỆC** | **NỘI DUNG CHI TIẾT CÔNG VIỆC** | **Nguyễn Gia Huy** | **Đinh Minh Thiện** |
| Tìm hiểu đề tài |  |  |  |
|  | Tìm hiểu Stack | X | X |
|  | Tìm hiểu winform C# | X | X |
|  | Thiết kế giao diện | X | X |
|  | Tìm hình ảnh cho game | X | X |
| Lảm đề tài |  |  |  |
|  | Xây dựng hàm trộn Shuffle |  | X |
|  | Xây dựng hàm Đổi chỗ Swap | X | X |
|  | Hàm Check win | X | X |
|  | Hàm Undo | X |  |
|  | Viết báo cáo |  | X |
|  | Powerpoint | X |  |